

Nama : Dandi Agung Setiawan
NIM : 1941720009
Kelas : TI-1F

Jawaban

2.3.3 Pertanyaan

1. Base line yang mencari nilai faktorial

```
public int faktorialDC(int n){  
    if(n == 1){  
        return 1;  
    }  
    else{  
        int fakto = n * faktorialDC(n-1);  
        return fakto;  
    }  
}
```

Base ini akan mencari nilai faktorial, Divide and Conquer lebih natural dengan skema rekursif yang dilakukan saat pencarian nilai faktorial dari program tersebut. Maksudnya jika nilai $n = 1$ maka akan di return jika salah akan masuk ke pencarian nilai fakto = n dikalikan (*) dengan faktorialDc ($n-1$) ; Dan akan di return nilai fakto-nya.

2. 3 bagian itu :

- a. Divide

```
if(n == 1){  
    return 1;  
}
```

- b. Conquer

```
else{  
    int fakto = n * faktorialDC(n-1);
```

- c. Combine

```
    return fakto;  
}
```

3. Bisa dapat diubah dengan while seperti berikut :

```
public int faktorialBF(int n){
    int fakto = 1;
    int i=1;
    while (i <= n) {
        fakto = fakto * i;
        i++;
    }
    return fakto;
    // int fakto = 1;
    // for(int i=1; i<=n; i++){
    //     fakto = fakto * i;
    // }
    // return fakto;
}
```

4. Code program

```
Faktorial[] fk = new Faktorial[elemen];
for(int i=0; i<elemen; i++){
    fk[i] = new Faktorial();
    System.out.print("Masukkan nilai data ke-" + (i+1) + " : ");
    fk[i].nilai = sc.nextInt();
}

System.out.println("=====");
long start = System.nanoTime();
System.out.println("Hasil faktorial dengan Brute Force");
for(int i=0; i<elemen; i++){
    System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialBF(fk[i].nilai));
}
long end = System.nanoTime();
System.out.println("Waktu eksekusi = " + (end-start) / 1000000 + " milisecond");
System.out.println("=====");
System.out.println("Hasil faktorial dengan Divide and Conquer");
for(int i=0; i<elemen; i++){
    System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialDC(fk[i].nilai));
}
System.out.println("Waktu eksekusi = " + (end-start) / 1000000 + " milisecond");
System.out.println("=====");
```

5. Hasil

```
=====
Masukkan jumlah elemen yang ingin dihitung : 25
Masukkan nilai data ke-1 : 1
Masukkan nilai data ke-2 : 2
Masukkan nilai data ke-3 : 3
Masukkan nilai data ke-4 : 4
Masukkan nilai data ke-5 : 5
Masukkan nilai data ke-6 : 6
Masukkan nilai data ke-7 : 7
Masukkan nilai data ke-8 : 8
Masukkan nilai data ke-9 : 9
Masukkan nilai data ke-10 : 10
Masukkan nilai data ke-11 : 11
Masukkan nilai data ke-12 : 12
Masukkan nilai data ke-13 : 13
Masukkan nilai data ke-14 : 14
Masukkan nilai data ke-15 : 15
Masukkan nilai data ke-16 : 16
Masukkan nilai data ke-17 : 17
Masukkan nilai data ke-18 : 18
Masukkan nilai data ke-19 : 19
Masukkan nilai data ke-20 : 20
Masukkan nilai data ke-21 : 21
Masukkan nilai data ke-22 : 22
Masukkan nilai data ke-23 : 23
Masukkan nilai data ke-24 : 24
Masukkan nilai data ke-25 : 25
=====
```

a. Menggunakan Brute Force

```
=====
Hasil faktorial dengan Brute Force
Faktorial dari nilai 1 adalah : 1
Faktorial dari nilai 2 adalah : 2
Faktorial dari nilai 3 adalah : 6
Faktorial dari nilai 4 adalah : 24
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 6 adalah : 720
Faktorial dari nilai 7 adalah : 5040
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 9 adalah : 362880
Faktorial dari nilai 10 adalah : 3628800
Faktorial dari nilai 11 adalah : 39916800
Faktorial dari nilai 12 adalah : 479001600
Faktorial dari nilai 13 adalah : 1932053504
Faktorial dari nilai 14 adalah : 1278945280
Faktorial dari nilai 15 adalah : 2004310016
Faktorial dari nilai 16 adalah : 2004189184
Faktorial dari nilai 17 adalah : -288522240
Faktorial dari nilai 18 adalah : -898433024
Faktorial dari nilai 19 adalah : 109641728
Faktorial dari nilai 20 adalah : -2102132736
Faktorial dari nilai 21 adalah : -1195114496
Faktorial dari nilai 22 adalah : -522715136
Faktorial dari nilai 23 adalah : 862453760
Faktorial dari nilai 24 adalah : -775946240
Faktorial dari nilai 25 adalah : 2076180480
Waktu eksekusi = 21 milisecond
=====
```

b. Menggunakan Divide and Conquer

```
=====
Hasil faktorial dengan Divide and Conquer
Faktorial dari nilai 1 adalah : 1
Faktorial dari nilai 2 adalah : 2
Faktorial dari nilai 3 adalah : 6
Faktorial dari nilai 4 adalah : 24
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 6 adalah : 720
Faktorial dari nilai 7 adalah : 5040
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 9 adalah : 362880
Faktorial dari nilai 10 adalah : 3628800
Faktorial dari nilai 11 adalah : 39916800
Faktorial dari nilai 12 adalah : 479001600
Faktorial dari nilai 13 adalah : 1932053504
Faktorial dari nilai 14 adalah : 1278945280
Faktorial dari nilai 15 adalah : 2004310016
Faktorial dari nilai 16 adalah : 2004189184
Faktorial dari nilai 17 adalah : -288522240
Faktorial dari nilai 18 adalah : -898433024
Faktorial dari nilai 19 adalah : 109641728
Faktorial dari nilai 20 adalah : -2102132736
Faktorial dari nilai 21 adalah : -1195114496
Faktorial dari nilai 22 adalah : -522715136
Faktorial dari nilai 23 adalah : 862453760
Faktorial dari nilai 24 adalah : -775946240
Faktorial dari nilai 25 adalah : 2076180480
Waktu eksekusi = 21 milisecond
=====
BUILD SUCCESSFUL (total time: 54 seconds)
```

2.4.3 Pertanyaan

1. Perbedaan 2 method PangkatBF() dan PangkatDC()

a. PangkatBF() menggunakan perulangan

```
public int pangkatBF(int a, int n){
    int hasil = 1;
    for(int i=0; i<n; i++){
        hasil = hasil * a;
    }
    return hasil;
}
```

b. PangkatDC() menggunakan fungsi rekursif

```
public int pangkatDC(int a, int n){
    if (n == 0){
        return 1;
    }
    else{
        if(n%2 == 1){ //Bilangan Ganjil
            return (pangkatDC(a,n/2) * pangkatDC(a,n/2) * a);
        }
        else{ //Bilangan Genap
            return (pangkatDC(a,n/2) * pangkatDC(a,n/2));
        }
    }
}
```

2. Potongan kode berikut :

```
if (n == 0){
    return 1;
}
else{
    if(n%2 == 1){ //Bilangan Ganjil
        return (pangkatDC(a,n/2) * pangkatDC(a,n/2) * a);
    }
    else{ //Bilangan Genap
        return (pangkatDC(a,n/2) * pangkatDC(a,n/2));
    }
}
```

Dalam kode program tersebut ada penyeleksian angka dimana angka n , jika $(n\%2) = 1$ maka n adalah bilangan ganjil dan akan dikembalikan ke nilai return berupa rekursif yaitu $(\text{pangkatDC}(a, n/2) * \text{PangkatDC}(a, n/2) * a)$ dan jika salah maka angka n merupakan bilangan genap sehingga akan mengembalikan nilai return rekursif pada $(\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2))$;

3. Tahap kombine sudah termaksud

```
return (pangkatDC(a,n/2) * pangkatDC(a,n/2));
```

4. Penambahan pada Class Pangkat

```
public int nilai, pangkat;

Pangkat() {
    this.nilai = nilai;
    this.pangkat = pangkat;
}
```

5. Penambahan pada Class PangkatMain

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Minggu3;
import java.util.Scanner;
/**
 * Nama    : Dandi Agung Setiaan
 * Kelas   : TI-1F
 * NIM     : 1941720009
 */
public class MainPangkat {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=====
        =====");
        System.out.println("          MENGHITUNG NILAI PANGKAT");

        System.out.println("=====
        =====");
        System.out.println("1. Brute Force");
        System.out.println("2. Divide and Conquer");

        System.out.println("=====
        =====");
        System.out.print("Masukkan pilihan Anda : ");
        int pilih = sc.nextInt();

        System.out.println("=====
        =====");
        System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
        int elemen = sc.nextInt();

        Pangkat[] png = new Pangkat[elemen];

        for(int i=0; i<elemen; i++){
            png[i] = new Pangkat();
            System.out.print("Masukkan nilai yang ingin dipangkatkan ke-" +
            (i+1) + " : ");
```

```

        png[i].nilai = sc.nextInt();
        System.out.print("Masukkan nilai pemangkat ke-" + (i+1) + " : ");
        png[i].pangkat = sc.nextInt();
    }
    switch(pilih) {
        case 1:

            System.out.println("=====
            =====");
                System.out.println("        Hasil Pangkat dengan Brute Force");

            System.out.println("=====
            =====");
                for(int i=0; i<elemen; i++){
                    System.out.println("Nilai " + png[i].nilai + " pangkat " +
                    png[i].pangkat + " adalah " + png[i].pangkatBF(png[i].nilai,
                    png[i].pangkat));
                    break;
                }

            case 2:

            System.out.println("=====
            =====");
                System.out.println("        Hasil Pangkat dengan Divide and
                Conquer");

            System.out.println("=====
            =====");
                for(int i=0; i<elemen; i++){
                    System.out.println("Nilai " + png[i].nilai + " pangkat " +
                    png[i].pangkat + " adalah " + png[i].pangkatDC(png[i].nilai,
                    png[i].pangkat));

            System.out.println("=====
            =====");
                    break;
                }

            default:
                System.out.println("Maaf pilihan yang Anda masukkan tidak
                valid!");
            }
        }
    }

```

```
} }
```

2.5.3 Pertanyaan

1. Perbedaan perhitungan

a) TotalBF() menggunakan for untuk melakukan perulangan dimana untuk menghitung keuntungan for(i=0; i<=elemen; i++) akan bertambah sampai terpenuhi yang akan dijabarkan dibawah ini :

- i=0 => total = 0 + keuntungan 1
- i=1 => total = (0 + keuntungan 1) + keuntungan 2
- i=2 => total = ((0 + keuntungan 1) + (keuntungan 2) + keuntungan elemen)

b) TotalDC() menggunakan Divide and Conquer

Misalkan bulan yang kita masukkan = 5, keuntungan = {1, 2, 3, 4, 5}.
Jadi perhitungan keuntungan dibagi menjadi 2 bagian kiri lsum dan bagian kanan rsum, yang akan saya jabarkan sebagai berikut :

- lsum = 1, 2 dan rsum = 4, 5 kemudian nilai mid (tengah) = 3
- di lsum melakukan perhitungan 1 + 2 = 3
- di rsum melakukan perhitungan 4 + 5 = 9

Kemudian direturn dengan menjumlahkan lsum + rsum + arr[mid] = 3 + 9 + 3 = 15 (hasil akhir)

2. Cara membatasinya dengan System.out.printf

```
System.out.printf("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalDC(sm.keuntungan, 0, sm.elemen-1));
```

3. Kode program tersebut adalah combine dari Divide and Conquer yaitu fungsi totalDC() itu sendiri, dimana fungsi digunakan untuk menggabungkan kembali pemecahan masalah tadi yang telah di conquer sebelumnya. Hasil akhir dari proses tersebut dan untuk menampilkan outputnya.

4. Variable mid berfungsi untuk mengetahui nilai tengah dari array arr[] yang dimana fungsi program mid ini yang mencari keuntungan perbulan yang didapatkan. Keuntungan perbulan yang didapatkan.

5. Kode program

```
/*  
 * To change this license header, choose License Headers in  
Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.
```



```

*/
package Minggu3;
import java.util.Scanner;
/**
 * Nama    : Dandi Agung Setiaa
 * Kelas   : TI-1F
 * NIM     : 1941720009
 */
public class MainSum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=====
        =====");
        System.out.println("  Progam Menghitung Keuntungan
        Total (Satuan Juta, Misal 5.9)");

        System.out.println("=====
        =====");
        System.out.print("Masukkan jumlah perusahaan : ");
        int elm = sc.nextInt();

        for(int i=0; i<elm; i++){

            System.out.println("=====
            =====");
            System.out.println("Perusahaan ke-" + (i+1));
            System.out.println("Masukkan jumlah bulan : ");
            int bln = sc.nextInt();
            }

            Sum sm = new Sum(elm);

            System.out.println("=====
            =====");
            for(int i=0; i<sm.elemen; i++){
                System.out.print("Masukkan untung bulan ke-" + (i+1)
                + " = ");
                sm.keuntungan[i] = sc.nextDouble();
            }

            System.out.println("=====
            =====");

```

```
System.out.println("          Algoritma Brute  
Force");
```

```
System.out.println("=====
```

```
System.out.println("Total keuntungan perusahaan  
selama " + sm.elemen + " bulan adalah = " +  
sm.totalBF(sm.keuntungan));
```

```
System.out.println("=====
```

```
System.out.println("          Algoritma Divide and  
Conquer");
```

```
System.out.println("=====
```

```
System.out.println("Total keuntungan perusahaan  
selama " + sm.elemen + " bulan adalah = " +  
sm.totalDC(sm.keuntungan, 0, sm.elemen-1));
```

```
}
```

```
}
```